Проект по сборке компактного клона ZX-80, в котором мы не только разберем сам процесс сборки, включая подготовку платы и схематику подключения на ней компонентов, но также познакомимся с используемым в ATX-80 интерпретатором Basic и научимся работать с памятью этого компьютера на примере загрузки 32 образцов программ.

ATX-80 – это наследник ZX-80, собранный на базе ATmega8. При этом он не эмулирует код, а заменяет оригинальный компьютер собственным кодом с аналогичной функциональностью на ассемблере AVR. Видео с него можно выводить как на VGA-монитор, так и на телевизор PAL/NTSC. Программы Basic можно хранить на внутренней/внешней EEPROM, внутренней флэш-памяти (32 слота), а также передавать между ПК и его хранилищем.

Получившийся клон отличается простотой сборки и компактностью. Его можно использовать как небольшой дешевый компьютер для обучения программированию на Basic. В общей сложности в него загружено 32 программы.

Общее описание проекта

Изначальной целью проекта было создание компьютера, сопоставимого по функционалу и техническим возможностям с оригинальным ZX-80. Это означало наличие полноценного интерпретатора Basic со всего 1Кб RAM и максимальное приближение к 4Кб ROM.

Еще одной задачей было изучить оригинальную ROM BIOS ZX-80, которая без преувеличения достойна похвалы. Помимо того, что она была инновационной, эта система создавалась с высокой оптимизацией размера кода, и несмотря на необходимость высокой ясности в потоке данных между функциями автор описал весь код с минимумом ошибок.

Я постарался сохранить функциональность исходного кода максимально точно и сопроводил его подробными комментариями. Невзирая на свой почтенный возраст, BIOS ZX-80 представляет очень полезный и информативный материал для изучения.

Для написания кода я выбрал процессор ATmega8, снаряженный 1Кб RAM (аналогично ZX-80) и 8Кб Flash. И хоть в оригинале объем флэш-памяти был вдвое меньше, переписанный код у меня получился где-то на 40% длиннее, поскольку инструкции AVR имеют размер 16 бит, плюс я добавил дополнительные функции, такие как более сложный генератор видео и обработку памяти, попутно задействовав 1Кб под хранение программ. В общем, 8Кб были распределены рационально.

ATX-80 оснащен клавиатурой с 40 микропереключателями. Он способен генерировать видео для VGA-монитора, а также выдавать сигнал в стандартах PAL и NTSC. Питание осуществляется от внешнего источника 5В через USB (USB-зарядки или ПК). Внешняя память EEPROM объемом 32Кб служит для хранения максимум 32 программ Basic в 32 слотах. Помимо этого, один внутренний слот флэш-памяти можно задействовать для копирования программ между внешними модулями EEPROM.

Редактируемая программа также автоматически отображается во внутреннюю EEPROM, размер которой составляет всего 512 байт, в связи с чем некоторые программы целиком в нее не входят, хотя большая часть все же имеет меньший размер (оперативная память разделяется между переменными и видеопамятью). В случае же с более объемными программами сохраняется, по крайней мере, их большая часть.

Преимущества ATX-80 перед оригинальным ZX-80:

- VGA-выход;
- изображение генерируется при выполнении программы (можно использовать режимы FAST и SLOW);
- редактируемая программа сохраняется даже после выключения;
- сохранение программ на внешнюю EEPROM;
- клавиши автоповтора;
- функция получения информации о памяти;
- повышенная скорость работы программ;
- меньший размер.

Ограничения ATX-80 в сравнении с ZX-80:

- невозможность выполнения программ в машинном коде (функция USR не работает);
- невозможность расширения хранилища внешней памятью;
- невозможность подключения дополнительных устройств через внешнюю шину;
- для хранения программ нельзя использовать кассетную пленку, доступна только EEPROM;
- сгенерированные символы содержат дополнительную 9-ю белую полоску (следствие генерации изображения через порт SPI)

Ошибки ZX-80 BIOS

В процессе изучения ZX-80 BIOS я обнаружил несколько мелких багов:

1) Ошибка команды REM. Адрес подпрограммы REM должен быть L0849, а не L084A. В результате этой ошибки выполнение REM без параметров (без текста) не приведет к выполнению следующей строки.

2) Ошибка в подпрограмме ED-EDIT – приведенные ниже инструкции должны идти в обратном порядке. Данная ошибка может привести к сбою системы, если выполнить команду при заполненной памяти:

```
JR NC,L03A7  ; если недостаточно места для заполнения ;строки редактирования, возвращение к ED-COPY

        POP HL                  ; восстановление позиции программы
```

3) еще одна ошибка, которую заметить сложнее, представляет риск при изменении компиляции. В начале подпрограммы L03CB ED-EDIT отсутствует инициализация регистра B на значение, представляющее количество строк для вывода (требуется значение 2 или больше). В этот момент регистр B содержит значение 3 (полученное из адреса L03CB), что вполне удовлетворительно, но переклмпиляция из другого адреса

может привести к прекращению вывода программы. Хотя я допускаю, что автор намеренно опустил здесь эту инструкцию в качестве элемента оптимизации.

Электрическая схема

В качестве основного процессора использован ATmega8. При желании его можно заменить на ATmega88, ATmega168 или ATmega328. В этом случае потребуется использовать другую прошивку, которая также прилагается в пакете для скачивания. Однако больший объем памяти старших процессоров в этом случае не задействуется.

Управляется ЦПУ тактовым генератором с частотой 24МГц. Такая частота необходима для отображения требуемого разрешения на VGA-мониторе. Несмотря на то, что у данного кристалла она выше пределов, заявленных разработчиком, на практике выяснилось, что с подобным разгоном справляются практически все процессоры.

Тем не менее рекомендуется использовать процессорный сокет, чтобы при возникновении сложностей можно было легко установить другую модель. Причиной неполадок также может выступить и недостаточно высокое напряжение. Частота процессора зависит от подаваемого питания, и низкое напряжение может вызвать его некорректную работу с используемым тактовым генератором.

Клавиатура оформлена в матрицу 5x8 клавиш по аналогичной ZX-80 схеме, только с обратным током. При ее опросе сигналы COL1 и COL5 поступают в качестве входа (то есть состояния высокого сопротивления), и всего один из сигналов COL установлен как выход с уровнем LOW.
Сигналы от ROW1 до ROW8 подключены в качестве входа с помощью внутреннего подтягивающего резистора. Когда происходит опрос, один из сигналов COL активируется на уровень LOW, и считывание сигналов ROW определяет нажатую клавишу.

Выход MOSI от генератора SPI направляется в качестве видеосигнала на VGA-коннектор и на Cinch-разъем, служащий для вывода изображения на ТВ. Предполагается, что этот выход будет нагружаться входным резистором 75Ом, который обеспечит необходимое напряжение в 0.7В. За вывод сигнала SYNC отвечает Timer1, обеспечивающий горизонтальную синхронизацию. Timer1 одновременно активирует прерывание для генератора видео, который отправляет видеоданные через контроллер SPI.
Сигнал VSYNC используется только в режиме старого VGA-дисплея. В других случаях на выходе SYNC обычно генерируется комбинированный синхросигнал CSYNC. Вертикальный сигнал создается в этой комбинации путем отрицания горизонтального синхросигнала.

Большинство современных VGA-мониторов поддерживают комбинированную синхронизацию и отлично работают в этом режиме. Только старые модели требуют дополнительного использования сигнала VSYNC. При этом на большинстве мониторов никакой разницы между режимами старых или новых моделей, как правило, не заметно.

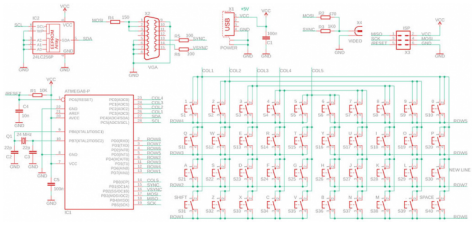
Внешняя память EEPROM подключена посредством интерфейса TWI/I2C (SDA и SCL). В качестве наиболее подходящего варианта рекомендуется использовать память 24LC256P 32Кб, которая может содержать до 32 программ (имеет 32 слота по 1Кб). Помимо этого, можно также использовать модули 24LC32 (4Кб, т.е. 4 слота), 24LC64 (8Кб, т.е. 8 слотов), 24LC128 (16Кб, т.е. 16 слотов) и 24LC512 (64Кб, т.е. 64 слота). Отличаются они только количеством доступных слотов.

Также рекомендуется использовать для EEPROM панельку, чтобы при необходимости можно было менять модули. Содержащиеся в памяти программы можно копировать между разными модулями как через внутренний слот флэш-памяти, так и с помощью прилагающейся утилиты.

В ZX-80 у модулей EEPROM есть входной контакт защиты от записи. В ATX-80 такой особенности нет, но для активации/отключения возможности записи можно добавить на плату перемычку.

Пара примечаний:
ATX-80 не распознает объем подключенной EEPROM, поэтому ответственность за использование корректного номера слота ложится на пользователя. Если вы задействуете номер слота, выходящий за диапазон доступной памяти, этот номер будет переопределен по принципу деления с остатком (EEPROM игнорирует неиспользуемые старшие адресные биты), что может привести к перезаписи поверх другой программы. Например, если в модуле памяти на 32Кб выполнить запись в слот 34, то по факту запись произойдет в слот 2.

Я думаю, что для запланированного клона ZX-81 использую аналогичное оборудование, только возьму другой процессор с большим объемом памяти (вероятно, Atmega328). Так что, советую установить панельку для удобства дальнейшей замены процессора.

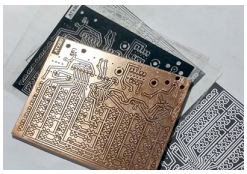


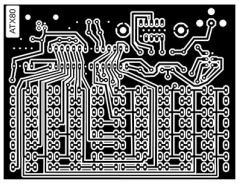
Сборка

ATX-80 я спроектировал под одностороннюю плату 100x75мм. Выбор пал на нее отчасти из-за ограничений бесплатной версии редактора Eagle, а отчасти из-за широкой доступности в таком размере печатных плат с фототравлением.

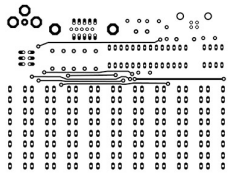
При изготовлении подобных плат я с помощью струйного принтера печатаю на прозрачной пленке фотошаблон (при этом склеиваю два для полноценного охвата), экспонирую их под УФ в течение 10 минут с расстояния 30см, проявляю в калийной щелочи и вытравливаю в хлорном железе.

Имейте ввиду, что в случае кастомизированного производства печатных плат необходимо проверять обозначения диаметров отверстий в шаблоне (я сверлил отверстия сам, поэтому не проверял).

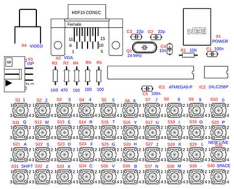




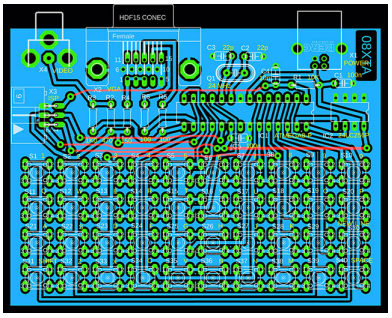
Нижняя часть платы



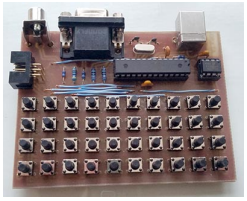
Верхняя часть платы, можно заменить перемычками из проводов



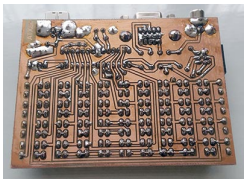
Шелкография

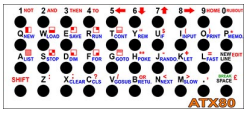


Компоновка элементов. Используйте панельки для процессора и памяти.



Плата в сборе





Наклейка с метками кнопок. Я напечатал ее на обычной бумаге также с помощью струйного принтера, смазал обе стороны клеем и сделал отверстия под кнопки размером 4мм

Полный комплект

Программирование и запуск

Установите FUSE-биты следующим образом:

- ATmega8: нижний 0xEF (по умолчанию 0xE1), верхний 0xC7 (по умолчанию 0xD9);
- ATmega88: нижний 0xF7 (по умолчанию 0x62), верхний 0xD7 (по умолчанию 0xDF), расширенный 0xFF (по умолчанию 0xF9);
- ATmega168: нижний 0xF7 (по умолчанию 0x62), верхний 0xD7 (по умолчанию 0xDF), расширенный 0xFF (по умолчанию 0xF9);
- ATmega328: нижний 0xF7 (по умолчанию 0x62), верхний 0xD7 (по умолчанию 0xD9), расширенный 0xFF (по умолчанию 0xFF).

Это делается для отображения загрузчика в наименьший размер памяти (необходимый для записи во внутреннюю флэш-память), защиты внутренней EEPROM во время программирования и увеличения скорости внешнего тактового генератора.

Если вы подключите VGA-монитор после программирования процессора, то должны увидеть простой белый экран с курсором “K” в нижнем левом углу. Если этого не произойдет, то, возможно, режим VGA не выбран. Для переключения режима нужно будет включить питание, зажав одну из клавиш от 1 до 4:

- 1 + включение питания: старый VGA-монитор (использует сигнал VSYNC);
- 2 + включение питания: новый VGA-монитор (использует комбинированную синхронизацию CSYNC);
- 3 + включение питания: режим NTSC TV (вывод через коннектор Cinch);
- 4 + включение питания: режим PAL TV (вывод через коннектор Cinch).

Выбранная конфигурация сохраняется в последнем байте EEPROM для дальнейшего использования. Если режим видео переключить не получается, это может говорить о коротком замыкании одной из кнопок.

Имейте ввиду, что нельзя использовать описанный выше способ для установки видеорежима на процессоре с помощью утилиты копирования. Это приведет к незапланированной активации программирования внешней EEPROM.

Проверить работоспособность процессора можно без подключения дисплея. Для этого нужно вставить в разъем VGA светодиод между контактами 14 (плюс, длинный) и 10

(короткий минус, короткий). Если вы включаете режим видео 1 (для старого монитора), то светодиод загорится (сигнал VSYNC активен). На включение других режимов он не отреагирует. В качестве альтернативы можете вставить светодиод в Cinch-разъем (плюс в центральное отверстие, минус к экрану) – в процессе генерации изображения светодиод будет гореть.

Программирование EEPROM

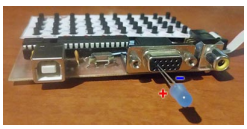
Загрузка программ в EEPROM

После тестирования базовой функциональности ATX-80 я советую запрограммировать внешнюю подключенную EEPROM. В прикрепленном архиве вы найдете набор утилит для загрузки 32 образцов программ. Для этого потребуется либо перепрограммировать основной процессор, либо использовать другой в качестве «копировальщика».

В наборе утилит есть каталог ATX80_Import, где лежат прошивки (BIN-файлы) для копирующего процессора, подготовленные под модели ATmega8/88/168/328 (первое число в имени файла указывает предполагаемую модель).

Для каждого типа процессора есть 3 набора программ: от слота 1 до 12, от слота 13 до 24 и от слота 25 до слота 32. Содержимое этих программ находится в том же пакете утилит в файлах Programs_1.txt, Programs_2.txt и Programs_3.txt. Описание программ также можно найти в файле Programs.txt из пакета исходного кода ATX-80.

Программа копирования с целью экономии памяти задействует только светодиодный индикатор, но не дисплей. Вставьте светодиод в VGA-разъем между контактами 14 и 10, как описано выше.



Загрузите файл *_1.bin с первым набором программ в процессор. Индикатор будет интенсивно мигать, указывая, что процессор находится в режиме ожидания. Если сейчас нажать 4, то светодиод перейдет в режим постоянного свечения, сообщая о выполнении программирования EEPROM. Затем он снова интенсивно замигает, указывая на завершение операции.

Загрузите остальные два набора программ аналогичным образом по очереди. Для регистрации нажатия кнопки она должна быть нажата не менее 0.5 секунды – светодиод должен светится не менее 1 секунды.

Копирование программ между EEPROM

Тот же пакет утилит содержит средства для скачивания или копирования программ из внешней EEPROM. Чтобы выполнить копирование из одной EEPROM в другую, следуйте описанной ниже процедуре.

Загрузите из каталога ATX80_Copy соответствующую программу в процессор. Мигающий светодиод будет сигнализировать о режиме готовности процессора. Нажмите одну из клавиш от 1 до 3. Светодиод загорится постоянно, указывая на выполнение операции, в результате которой программы из EEPROM будут скачаны во внутреннюю флэш-память процессора. Выбранная клавиша определяет скачиваемые слоты: 1 выбирает слоты с 1 по 11, 2 выбирает слоты с 12 по 22, а 3 выбирает слоты с 23 по 32.

В теории ATmega8 не способен вместить 11 слотов – это бы означало 11Кб памяти при имеющихся у этой модели 8Кб. Если в некоторых случаях программы входить не будут, то вам потребуется использовать более вместительный процессор или скорректировать количество слотов в исходном коде программы копирования.

После загрузки программ в процессор выключите питание, замените модуль EEPROM и снова его включите. Нажмите 4 для сохранения загруженных программ из флэш-памяти в новую внешнюю EEPROM. Все находящиеся в памяти программы сохраняются в те же слоты, из которых копировались.

Скачивание программ из EEPROM

Вы также можете скачивать программы на ваш ПК с помощью утилит. Для начала, как и в прошлом случае, нужно загрузить подходящий набор программ во флэш-память процессора, используя клавиши 1-3. Далее подключите программатор ЦПУ и скачайте содержимое флэш-памяти в каталог ATX80_Copy в файлы 1.bin и 3.bin. В каталоге ATX80_Export запустите командный файл !.bat. Так вы экспортируете программы из скачанных образов флэш-памяти в текстовые файлы Programs_1.txt, Programs_2.txt и Programs_3.txt, находящиеся в базовом каталоге утилит.

Содержимое этих файлов можно отредактировать в текстовом виде. В начале кода каждой программы прописана метка Slot x, сообщающая, к какому слоту она принадлежит. Эта метка указывает начало программы для дальнейшей обработки. Все последующие строки, начинающиеся с числа, обрабатываются как строки программы. Эти строки могут сопровождаться инструкциями LET или DIM, содержащими заполнение для численных или текстовых переменных.

Если вы хотите загрузить программы в текстовом виде обратно в EEPROM, то запустите файл !.bat в каталоге ATX80_Import. Это обеспечит импорт всех текстовых форм программ обратно в образы программы копирования. Далее вам останется лишь загрузить эти программы в процессор и нажать 4, чтобы сохранить их во внешней EEPROM аналогично тому, как это описывалось в начале текущего раздела.
Описание ATX-80 BASIC

Basic на ATX-80 почти полностью идентичен Basic с ZX-80, за исключением нескольких небольших отличий.

Номер строки программы может простираться от 1 до 9999. После каждого изменения в строках программы во внутренней EEPROM автоматически создается ее резервная

копия. Емкость этой EEPROM составляет всего 510 байт, поэтому целиком программы в нее вмещаются не всегда. При включении ATX-80 программа, сохраненная во внутренней памяти, восстанавливается.

Basic на ATX-80 работает с целыми числами в диапазоне от -32768 до +32767.

Типы переменных:

- целочисленная переменная с произвольно длинным именем (начинается с буквы, которая может сопровождаться буквами или цифрами);
- целочисленная переменная со всего одной буквой может использоваться в качестве параметра цикла FOR;
- текстовая переменная – 1 буква, сопровождаемая $. Текст может иметь любую длину;
- массив представляется одной буквой и ограничен размером в 255 элементов.

Операции

- + — … сумма и разность;
- * /… умножение и деление (только для целых чисел);
- AND… побитовое логическое произведение;
- OR… побитовая логическая сумма;
- NOT… побитовое логическое «не»;
- **… степень
- =… проверка равенства чисел или текстов;
- <… сравнение чисел и текстов с меньшим;
- >… сравнение чисел и текстов с большим.

Команды

Могут идти только в начале строки или после THEN и вызываются нажатием соответствующей кнопки:

- LIST … (A) вывод листинга программы с заданной строки. Если номер строки не указывать, выводится листинг всей программы;
- RETURN … (B) возвращение из процедуры;
- CLS ...(C) очистка экрана;
- DIM … (D) объявление массива. Массив должен начинаться с индекса 0 и может содержать до 255 элементов (например, при DIM A(255) в нем могут быть задействованы индексы от 0 до 255);
- SAVE … (E) сохранение программы и переменных в слот. Эта команда сопровождается номером слота внешней EEPROM от 1 до 32. Если число не указать или указать 0, программа и переменные сохранятся во внутренней флэш-памяти процессора.
- FOR … (F) цикл, сопровождаемый однобуквенной переменной с начальным значением, после которой идет TO с конечным значением;
- GO TO… (G) переход к указанной строке;
- POKE … (H) сохранение байта в RAM, сопровождаемое адресом содержимого этого байта. ATX-80 имитирует сохранение в адресах RAM идентично ZX-80;
- INPUT … (I) пользовательский ввод, сопровождаемый целевой численной или текстовой переменной;

- RANDOMISE … (J) инициализация генератора случайных чисел от указанного числа. Если ввести 0 или опустить параметр, инициализируется согласно счетчику фреймов;
- LET … (K) установка содержимого переменной;
- FAST … (L) установка ускоренного режима. Отображается только одна верхняя строка, и процессор ускоряется в 4 раза. Использовать можно только при выполнении программы;
- SLOW … (M) установка медленного режима. Отображается весь экран. Включается автоматически при выходе из программы.
- NEXT … (N) продолжение цикла для заданной переменной;
- PRINT … (O) вывод переменных или текста. Символ ; обеспечивает продолжение без пробела. Символ , выставляется после 8-символьных позиций табуляции;
- MEMORY … (P) вывод информации о памяти в байтах: размер программы, переменных, а также свободное пространство RAM;
- NEW … (Q) удаление программы и переменных;
- RUN ...® выполнение программы с заданной строки. Если номер строки не указать, программа выполняется с начала. В то же время происходит очистка переменных.
- STOP … (S) остановка программы. Для продолжения используется CONTINUE.
- CONTINUE … (T) возобновляет выполнение программы, остановленной командой STOP.
- IF … (U) условие. THEN сопровождается инструкцией, выполняемой при валидности условия или выражения (выражение считается валидным, если не является нулевым);
- GO SUB… (V) переход к процедуре. Из процедуры инструкция RETURN возвращает к строке, следующей за GO SUB.
- LOAD … (W) загрузка программы и переменных из слота. Эта команда сопровождается значением от 1 до 32 для внешней EEPROM. Если значение не ввести или ввести 0, программа и переменные будут загружены из внутренней флэш-памяти процессора;
- CLEAR … (X) очистка переменных;
- REM … (Y) комментарий.

Функции и прочие команды

функции необходимо прописывать, так как клавиатурных сокращений для них нет:

- "… (Shift+Y) указывает начало и конец текста;
- THEN … (Shift+3) часть условия IF, указывающая команду, которая выполняется при валидности условия;
- TO … (Shift+4) часть цикла FOR, указывающая конечное значение переменной;
- ;… (Shift+Z) часть инструкции PRINT, продолжающая вывод с последней позиции;
- ,… (Shift+.) часть инструкции PRINT, продолжающая вывод из позиции в таблице;
- ( )… (Shift+I, Shift+O) скобки;

- PEEK … загружает содержимое памяти из адреса, указанного в скобках. ATX-80 имитирует адреса RAM идентично ZX-80, включая адрес шрифта в ROM. Содержимое указателей не имитируется, их необходимо вычислять повторно;
- CHR$… преобразует символьный код в текстовый символ;
- CODE … преобразует текстовый символ (первый символ текста) в численный код;
- RND … генератор случайных чисел. Генерирует число в диапазоне от 1 до заданного значения;
- TL$… возвращает текст без первого символа;
- USR … в ATX-80 не поддерживается;

STR$… преобразует число в текст;

- ABS … абсолютное значение числа.

Примеры программ

Эти программы хранятся во внешней EEPROM в слотах 1 – 32. Содержимое EEPROM можно скопировать в другую EEPROM, скачать на ПК, либо загрузить с ПК в EEPROM, как описывалось в разделе «Программирование EEPROM».

Содержимое программ вместе с подробным описанием находится в исходном коде в файле Programs.txt.

VELOC HEIGHT FUEL
-80   1500   7000

COST 10.9919

0/510

ENTER ROW AND HOW MANY
13

1 2 0
6 X 4
7 6 5

I MOVED
1 2 0
6 X X
7 6 5

YOUR MOVE

YOURS   BTSRO
ET 0   T J
ET 490

1333 -22 -4 7 22 45 344 344 655
699

0/270

```
   10 REM LINE RENUMBER
   20 GO SUB 1000
   30 GO SUB 1987
9999 POKE 16368,0
9999 POKE 16368,0
9999 POKE 16368,0
9999 IF PEEK(197)=191 THEN 17400
9999 IF PEEK(197)=251 THEN PRINT L
9999 OPEN #252,
9999 IF PEEK(N)=236 THEN PRINT L
9999 CLOSE #252
9999 IF PEEK(N)=118 THEN GO
TO 9999
9999 IF PEEK(N)=126 THEN STOP
9999 POKE 561,1:REM
9999 POKE 1481,PEEK AND 255
9999 POKE 53774,
9999 GET #1,N
9999 NEXT N
```

0

```
SOUND
1
```

```
   32449
ENTER CODE?
```

I CALCULATE DAY YOU WERE BORN
PLEASE ENTER YOUR NAME
HELLO MIKE!
ENTER DATE OF BIRTH:
DAY
MONTH
YEAR
MIKE WAS BORN ON SUN
ANYONE ELSE? (Y)

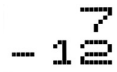
THINK OF A NUMBER BETWEEN 1 AND
310
PRESS <CR> WHEN READY
NOW DIVIDE YOUR ORIGINAL NUMBER
BY 3
PLEASE ENTER REMAINDER
NOW DIVIDE YOUR ORIGINAL NUMBER
BY 5
PLEASE ENTER REMAINDER
NOW DIVIDE YOUR ORIGINAL NUMBER
BY 7
PLEASE ENTER REMAINDER
YOUR NUMBER WAS 221
ANOTHER? (Y)

NEXT CHMM IS

PRINT ENTIRE SEQUENCE
?2048

HELL!
BBBBBB

TOO BAD
ENTER NEW WORD



WHAT IS THE CAPITAL OF
HOLLAND TEY CHAIN
? AMSTERDAM
WHAT IS THE CAPITAL OF
FRANCE
?

PRESS R/L

SOLUTIONS TO 2 EQUATIONS:
AIX + BXY + C = 0
ENTER DATA
A 5
B 7
C -90
SOLUTIONS ARE
-1 3

9/640

Материалы для скачивания

- Схема в Eagle Free ATX80_diagram.zip
- Графика ATX80_png.zip
- Исходный код (скомпилирован через WinAVR для ATmega8, ATmega88, ATmega168 и ATmega328) ATX80_src.zip
- Утилиты для загрузки и скачивания примеров программ в EEPROM, скомпилированы в MSVC 2005 ATX80_util.zip

Список деталей

- 432-027, процессор ATmega8A-PU, 1 шт. Также подойдут ATmega88 (более дешевый), ATmega168 или ATmega328 в корпусе PDIP-28.
- 414-041, 32Кб (Кбит) EEPROM 24LC256-I/P, 1 шт.
- 131-090, тактовый генератор 24МГц HC49U/S, 1 шт.
- 120-060, керамический конденсатор 100нФ, 2 шт.
- 120-251, керамический конденсатор 10нФ, 1 шт. Емкость большой роли не играет.
- 120-125, керамический конденсатор 22пФ, 2 шт. Можно взять 27пФ.
- 114-453, резистор 100Ом, 2 шт. Допустимо заменить на меньшую емкость (с большей емкостью лучше не использовать – можно нарушить синхронизацию).
- 114-455, резистор 150Ом, 1 шт.
- 114-460, резистор 470Ом, 1 шт.
- 114-463, резистор 1кОм, 1 шт.
- 114-474, резистор 10кОм, 1 шт. Емкость большой роли не играет.
- 823-011, DIP-панелька на 8 контактов SOKL8, 1 шт.
- 823-026, DIP-панелька на 28 контактов SOKL 28U, 1 шт.
- 630-749, микропереключатели для платы ТС-0107-Т, 40 шт.
- 832-120, USB 2.0 B для платы 90° DS1099-WN0, 1 шт.

- 801-003, D-Sub 15-контактный сокет для платы 90° CAN 15 3Z 90, 1 шт.
- 808-166, гнездо RCA для платы 90° RS-101F, 1 шт.
- 821-256, вилочный разъем 2x3 контакта для платы 90° MLW06A, 1 шт.
- 661-064, однослойная плата с нанесенным фоторезистом 75x100x1.5мм, 1 шт.